**UNIVERSIDAD PONTIFICIA BOLIVARIA SECCIONAL BUCARAMANGA**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**LÓGICA DE PROGRAMACIÓN**

**JUAN JOSÉ DÍAZ PRADA**

# **Taller colaborativo de funciones**

🧐

Objetivos:

* El estudiante estará en capacidad aplicar los conceptos estudiados en clase sobre el uso de las funciones para resolver problemas con el lenguaje Python.
* El estudiante participará activamente en clase en la solución de los problemas planteados, compartiendo código, explicaciones y opiniones sobre las soluciones propuestas en la clase.
* El estudiante trabajará organizadamente todo su código fuente en su repositorio de GitHub y comentará este código fuente.

**Ejercicios.**

Cuando resuelvas los ejercicios estarás actuando como el “programador”, o sea la persona que diseña y escribe el programa en este caso en Python. Pero luego tendrás que ejecutar tu código y probar si funciona correctamente, y en este caso actuarás como el “usuario”, o sea alguien que utiliza el programa.

No siempre el programador y el usuario son la misma persona y, generalmente, los usuarios no conocen los detalles técnicos ni pueden ver el código, solo verán la salida en pantalla que les muestre el programa.

Considera entonces que siempre se debe pensar en términos de usabilidad, es decir que los mensajes para el usuario sean claros y le permitan ingresar los datos que en realidad tú necesitas como programador.

Se debe procurar que nuestro programa funcione de forma consistente y sin errores en todos los casos.

A continuación, se pide desarrollar una serie de ejercicios con funciones, puede que exista otra manera de hacerlo, pero en este caso, se piden con funciones (def – definidas por el usuario).

Lo que sí puedes hacer es mejorar el código propuesto y dar otras alternativas de solución optimizando el uso de funciones.

Compartir hace que todos aprendamos.

1. Solicitar al usuario que ingrese su dirección de correo electrónico. Imprimir un mensaje indicando si la dirección es válida o no, valiéndose de una función para decidirlo. Una dirección se considerará válida si contiene el símbolo "@".

def validar(email):

caracterBuscado="@"

emailValido=False

for c in email:

if c==caracterBuscado:

return True

return False

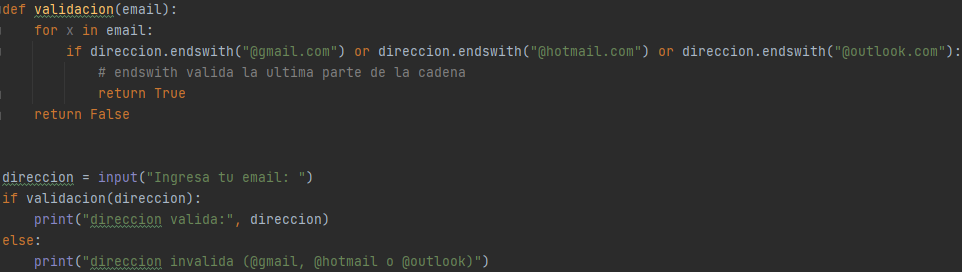
direccion=input("Tu email: ")

if validar(direccion):

print("Dirección válida")

else:

print("Dirección inválida")





Código original por Edwin Nicolás Tarazona, eliminadas algunas variables innecesarias y la salida se hizo mas fácil de leer para el usuario

1. Solicitar números al usuario hasta que ingrese el cero. Por cada uno, mostrar la suma de sus dígitos (utilizando una función que realice dicha suma).

def sumaDigitos(numero):

    suma=0

    while numero!=0:

        digito=numero%10

        suma=suma+digito

        numero=numero//10

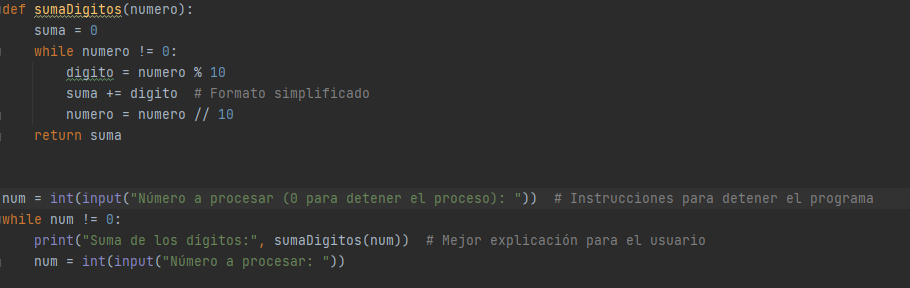
    return suma

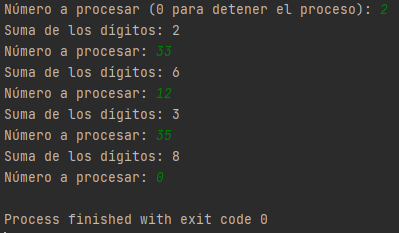
num=int(input("Número a procesar: "))

while num!=0:

print("Suma:",sumaDigitos(num))

num=int(input("Número a procesar: "))





1. Solicitar números al usuario hasta que ingrese el cero. Por cada uno, mostrar la suma de sus dígitos. Al finalizar, mostrar la sumatoria de todos los números ingresados y la suma de sus dígitos. Reutilizar la misma función realizada en el ejercicio 2.

def sumaDigitos(numero):

    suma=0

    while numero!=0:

        digito=numero%10

        suma=suma+digito

        numero=numero//10

    return suma

sumatoria=0

num=int(input("Número a procesar: "))

while num!=0:

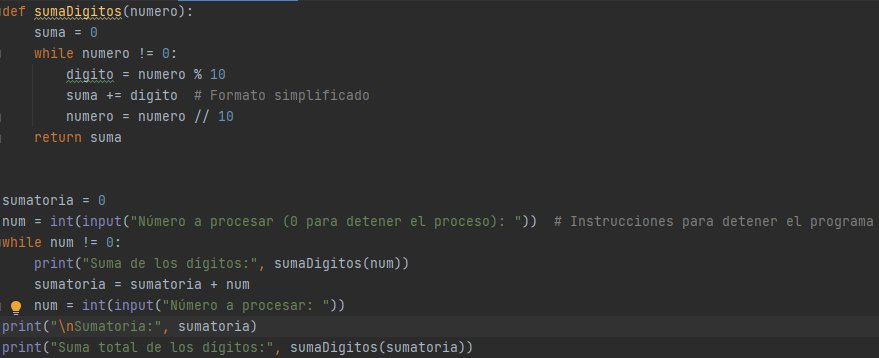
print("Suma:",sumaDigitos(num))

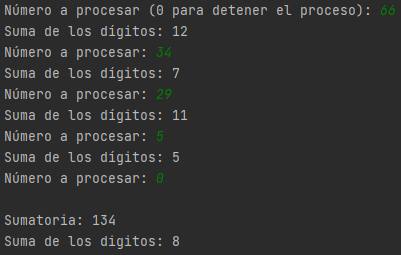
sumatoria=sumatoria+num

num=int(input("Número a procesar: "))

print("Sumatoria:", sumatoria)

print("Dígitos:", sumaDigitos(sumatoria))





1. Solicitar al usuario que ingrese un número entero e informarle si es primo o no, utiliza en el código una función booleana que lo decida.

def primo(num):

for i in range(2,num):

if num%i==0:

return False

return True

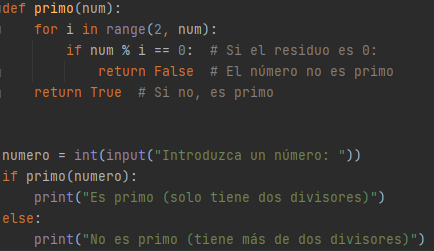
numero=int(input("Número: "))

if primo(numero):

print("Es primo")

else:

print("No es primo")







1. Solicitar al usuario un número entero y luego un dígito. Informar la cantidad de ocurrencias del dígito en el número, utiliza para ello una función que calcule la frecuencia del dígito en el número ingresado.

def frecuencia(numero,digito):

cantidad=0

while numero!=0:

ultDigito=numero%10

if ultDigito==digito:

cantidad+=1

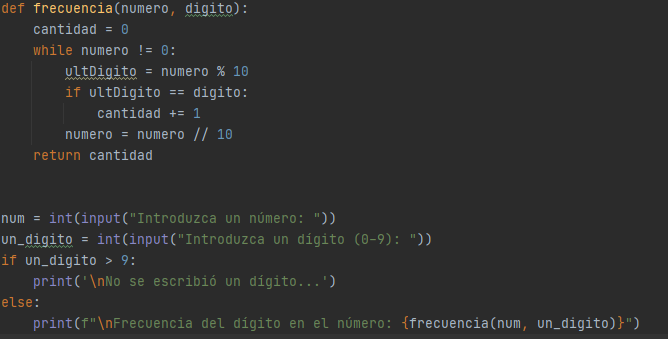
numero=numero//10

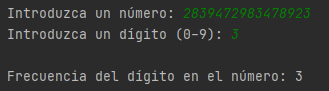
return cantidad

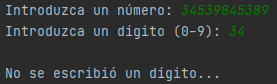
num=int(input("Número: "))

un\_digito=int(input("Dígito: "))

print("Frecuencia del dígito en el número:",frecuencia(num,un\_digito))







1. Escribir un programa que pida números al usuario, mostrar el factorial de cada uno y, al finalizar, la cantidad total de números leídos en total. Utilizar una o más funciones, según sea necesario.

def factorial(numero):

f=1

if numero!=0:

for i in range(1,numero+1):

f=f\*i

return f

cantidad=0

num=int(input("Número (-1 para cortar): "))

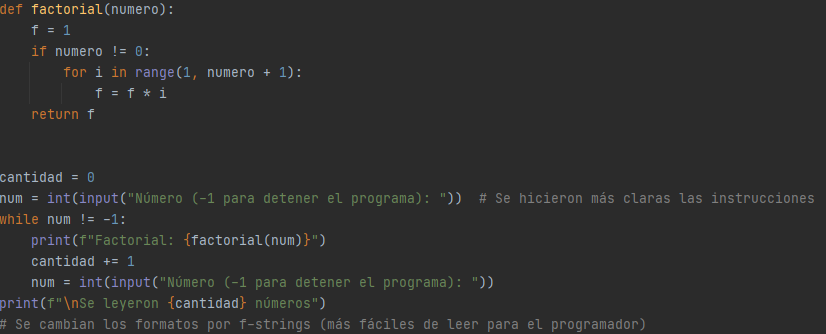
while num!=-1:

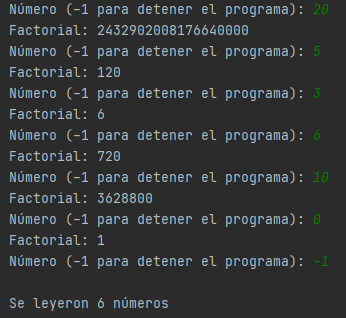
print("Factorial:", factorial(num))

cantidad+=1

num=int(input("Número (-1 para cortar): "))

print("Se leyeron",cantidad,"números")





1. Escribir un programa que pida números positivos al usuario. Mostrar el número cuya sumatoria de dígitos fue mayor y la cantidad de números cuya sumatoria de dígitos fue menor que 10. Utilizar una o más funciones, según sea necesario.

def sumaDigitos(numero):

suma=0

while numero!=0:

digito=numero%10

suma=suma+digito

numero=numero//10

return suma

cantidad=0

mayor=-1

numero=int(input("Número positivo: "))

while numero>=0:

suma=sumaDigitos(numero)

if suma > mayor:

mayor=suma

n\_mayorsuma=numero

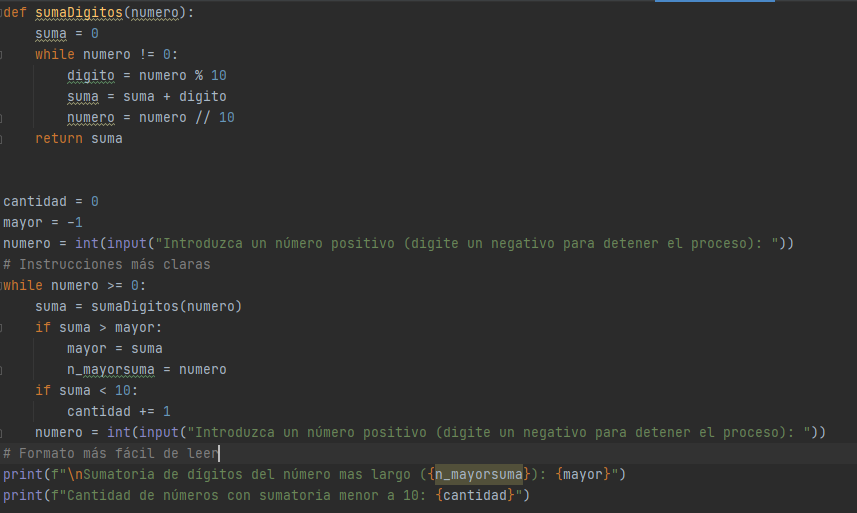
if suma < 10:

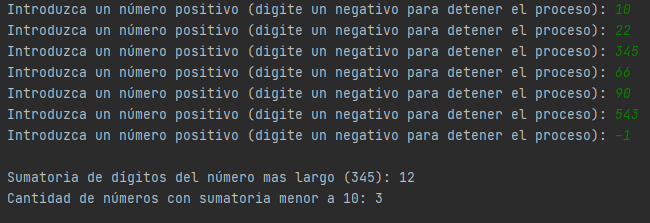
cantidad+=1

numero=int(input("Número positivo: "))

print("Sumatoria de dígitos de",n\_mayorsuma,":",mayor)

print("Cantidad con sumatoria menor a 10:",cantidad)





1. Solicitar al usuario el ingreso de números primos. La lectura finalizará cuando ingrese un número que no sea primo. Por cada número, mostrar la suma de sus dígitos. También solicitar al usuario un dígito e informar la cantidad de veces que aparece en el número (frecuencia). Al finalizar el programa, mostrar el factorial del mayor número ingresado.

def primo(num):

for i in range(2,num):

if num%i==0:

return False

return True

def frecuencia(numero,digito):

cantidad=0

while numero!=0:

ultDigito=numero%10

if ultDigito==digito:

cantidad+=1

numero=numero//10

return cantidad

def factorial(numero):

f=1

if numero!=0:

for i in range(1,numero+1):

f=f\*i

return f

def sumaDigitos(numero):

suma=0

while numero!=0:

digito=numero%10

suma=suma+digito

numero=numero//10

return suma

mayor=0

numero=int(input("Número primo: "))

while primo(numero):

print("Suma de los dígitos:",sumaDigitos(numero))

digito=int(input("Dígito: "))

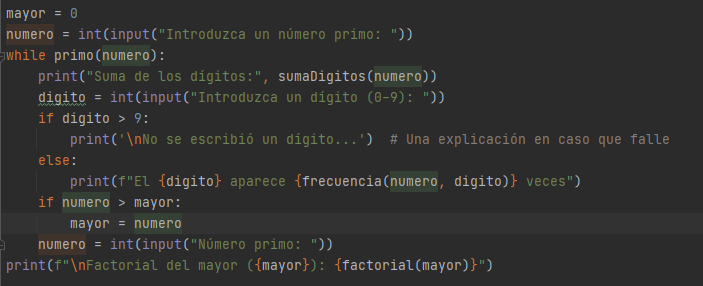
print("El",digito,"aparece",frecuencia(numero,digito),"veces")

if numero > mayor:

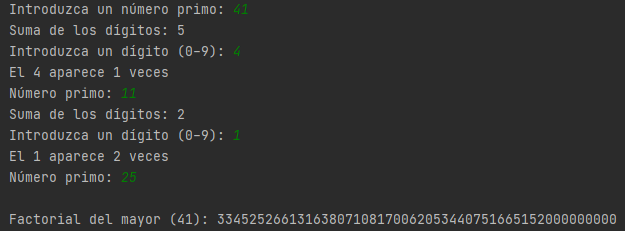
mayor=numero

numero=int(input("Número primo: "))

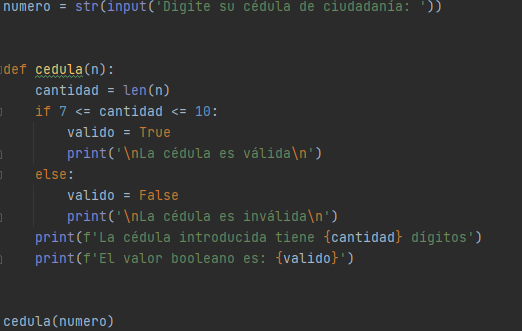
print("Factorial de",mayor,":",factorial(mayor))

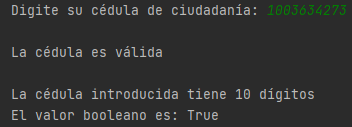


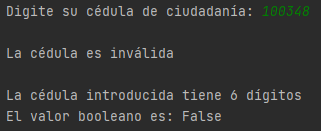
(Parte modificada)



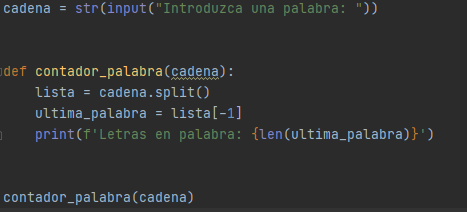
1. Escribir una función que, dado un número de cédula de ciudadanía, retorne True si el número es válido y False si no lo es. Consulta cuál es la longitud válida para cédula en Colombia.







1. Escribir una función que, dada una cadena, retorne la longitud de la última palabra. Se considera que las palabras están separadas por uno o más espacios. También podría haber espacios al principio o al final del string pasado por parámetro. Consulta sobre la función len() en Python.





Código original por Edwin Nicolás Tarazona, modificado para ser mas amigable con el usuario